LANGUES NATURELLES ET ARTIFICIELLES ET TRADUCTION AUTOMATIQUE

Résumé de la conférence prononcée par monsieur J. A. Baudot, le 15 février 1967, à l'Université de Montréal.

par Pierre Bouchard, étudiant à l'Université de Montréal

Lorsqu'on a commencé à programmer directement les premières calculatrices en langage machine, ce travail s'est révélé fort fastidieux car il nécessitait un grand nombre d'opérations pour exprimer des concepts fort simples. Aussi a-t-on inventé des langages algorithmiques, plus près de la langue parlée et du langage mathématique: il devenait alors plus simple de rédiger les programmes en langage algorithmique et la traduction du langage algorithmique au langage machine devenait ensuite une opération de routine ... si bien qu'on a vite programmé les ordinateurs pour qu'ils fassent cette traduction eux-mêmes: ainsi, au centre de calcul de l'Université de Montréal, on communique le programme à la machine le plus souvent dans le langage FORTRAN et elle traduit les énoncés de ce langage en énoncés (généralement beaucoup plus nombreux) de langage machine qu'elle exécute ensuite.

Les calculatrices pouvant faire ce genre de traduction, ne seraient-elles pas capables de traduire une langue naturelle en une autre? Pour avoir une idée des problèmes que pose une telle traduction automatique, voyons d'abord quelques notions mathématiques sur les langues et les grammaires.

Pour définir un langage mathématique, on se donne un ensemble fini V d'éléments appelés mots. A partir de cet ensemble V, on forme un nouvel ensemble V* qui est l'ensemble des suites finies d'éléments de V. Par définition, un langage L sur V* est un sous-ensemble L de V*. Par exemple, soit V l'ensemble des drapeaux utilisés dans la marine pour communiquer d'un navire à l'autre; les suites finies de drapeaux utilisées par les marins sont les éléments de ce langage. Les éléments de V* sont appelés phrases.

Pour déterminer un langage de V*, on pose habituellement un ensemble de règles qui constituent une grammaire G. Le langage formé à partir de ces règles est le langage engendré par G et noté L (G). Voici des exemples:

Exemple I: Une des grammaires les mieux connues: grammaire C.F.

$G = (Vn, Vt, S, P)$

où $Vn = \{A, B, \ldots\}$ est un ensemble appelé vocabulaire auxiliaire

$Vt = \{a, b, \ldots\}$ est appelé vocabulaire terminal

$Vn \cap Vt = \emptyset$

$Vn \cup Vt = V$ est le vocabulaire de la langue engendrée par G

S est un élément initial du vocabulaire auxiliaire

$P = \{A \rightarrow \varphi\}$ est un ensemble de règles de la forme $A \rightarrow \varphi$ (lire "A peut être réécrit comme φ ") où $A \in Vn$, $\varphi \in V^*$ (formellement, P est un ensemble de couples (A, φ)); P est appelé ensemble des productions.

Exemple I a:

G = (Vn, Vt, S, P)
Vn = {S, A, B}
Vt = {a, b, c}
P = {S → A, A → aAb, A → cB, B → b}

On a ici L(G) = $\{a^n c\, b^{n+1}\}$ $n \geqslant 0$

(ici a^n est une abréviation pour a a ... a, n fois)

Voici, par exemple, comment on obtient a c b b à partir des règles

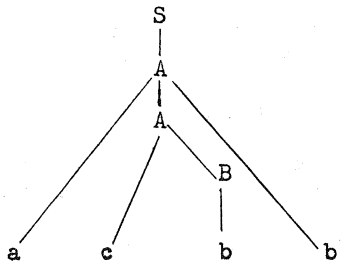
S → A

A → a A b

a A b → a c B b

a c B b → a c b b

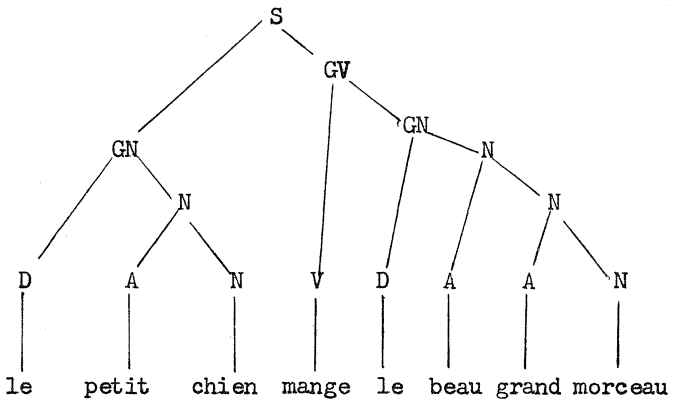
ceci est habituellement représenté par un arbre:



Exemple I b:

G = (Vn, Vt, S, P)
Vn = {S, N, D, V, A, GN, GV}
Vt = {le, petit, grand, morceau, beau, mange, chien}
P = {S → GN, GV N.B. (S → GN, GV est une abréviation pour "S → GN, S → GV")
GN → D, N
GV → V, GN
N → A, N
D → le
A → petit, grand, beau
N → chien, morceau
V → mange}

Voici une phrase engendrée par cette grammaire ainsi que l'arbre qui montre la manière dont elle a été obtenue:

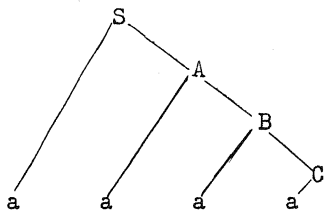


On voit qu'ici le vocabulaire initial contient les désinences de catégories de mots V: verbe, GN: groupe nominal, etc. On observe également que la règle $N \rightarrow A$, N est récursive: le langage engendré par G contient une infinité de phrases. Mais on remarque aussi que L(G) contient des phrases comme "le petit grand beau grand grand chien" que l'usage correct condamme.

Exemple II: Un autre type de grammaire: grammaire F.S.

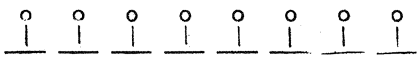
G = (Vn, Vt, S, P) comme précédemment, mais on ajoute la restriction que les éléments $A \rightarrow \varphi$ de P soient avec $\varphi = aB$ ou $\varphi = a$ ($a \in Vt$, $B \in Vn$).

Voici un exemple de phrase engendrée par une grammaire F.S.



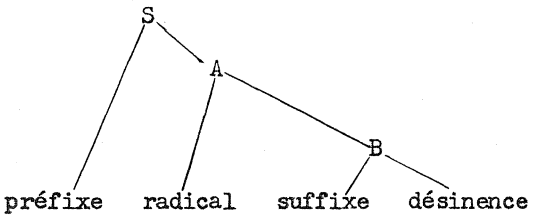
Cet exemple nous amène à la traduction automatique. Notre grammaire doit remplir trois conditions: elle doit 1^{o}, définir une langue, 2^{o}, engendrer toutes les phrases de cette langue et 3^{o}, nous permettre de reconnaître si une phrase quelconque appartient à la langue en question. Bien qu'on définisse souvent une grammaire sous son aspect générateur, ce troisième critere, celui de la reconnaissance, est le plus important; il y a cependant des grammaires plus compliquées que celles-ci où il est impossible de décider si une phrase donnée appartient à L(G).

Dans une langue naturelle, l'analyse grammaticale d'une phrase se fait en deux étapes: d'abord une analyse morphologique qui a pour but de reconnaître la catégorie (nom, adjectif...) à laquelle appartient chaque mot (bas de l'arbre:



les traits indiquant les mots, les o leur catégorie), ensuite une analyse syntaxique destinée à reconnaître la structure de la phrase.

On peut procéder de deux façons pour faire l'analyse morphologique. La première consiste à compiler un dictionnaire de tous les mots et de toutes les formes possibles de ceux-ci; cette méthode est pratique pour l'anglais, par exemple, qui ne compte pas trop de formes, mais pour les langues marquées, comme le français ou le russe avec les déclinaisons, on procède autrement: on utilise une grammaire FS (FS pour Finite State) et on analyse les différentes parties du mot (cf. arbre ci-dessous).



ex: verte

radical désinence du féminin

INUTILITES

préf. rad. suf. dés.

Dans cette grammaire FS, une phrase est un mot du français et la langue est l'ensemble de tous les mots.

Mais, déjà au niveau de l'analyse morphologique se pose le problème des homographes: ainsi le mot PARTIES peut être un substantif ou le participe passé de partir et l'ordinateur qui en a mémoire

sous la rubrique:

radical		désinence
PARTI	--	ES
PART	--	IES

va pouvoir décomposer le mot de deux façons. Il faut donc un grand nombre de catégories et de règles pour reconnaître les découpures parasites.

Ensuite, après avoir trouvé la (ou les) catégorie (s) de chaque mot, on doit découvrir la structure qui unit les éléments (mots). Là encore se pose le problème des homographes. Il y a des homographies naturelles comme

"L'homme brave la tourmente"

(où le sens est tout autre selon que brave soit verbe ou adjectif, mais où les deux analyses sont correctes en français) ou encore comme

"Time flies like an arrow"

(ou on peut considérer time, flies ou like comme verbe, la phrase gardant un sens dans chaque cas et l'analyse syntaxique étant correcte aussi). Il y a aussi des homographies parasites: par exemple,

"Le chat mange la souris" (souris: complément d'object direct)
"Le hibou mange la nuit" (nuit: complément circonstanciel)

la seconde phrase admet aussi la même analyse que la première, mais il ne s'agit plus alors d'une phrase française (1) car on ne peut manger "la nuit" qui est un concept abstrait. Voici une autre homographie parasite:

"I waited for an hour"
"I waited for a bus".

Les tentatives actuelles pour résoudre ce problème consistent à introduire des traits sémantiques qu'on "attache" aux mots du vocabulaire: ainsi nuit - abstrait souris - concret. D'ailleurs le principal problème est celui de la reconnaissance: l'ordinateur reconnaît dans la langue des phrases qui n'en sont pas, car il utilise une grammaire CF (Context Free) qui ne tient pas compte du contexte qui pourrait aider à éliminer les structures parasites.

Un autre problème qui se pose est celui des éléments discontinus. (En français: ne ... pas; en anglais: I give up the conrest, I give it up; en allemand: Ich setze das Buch über). On trouve présentement moyen de regrouper ces éléments en introduisant des contraintes dans les grammaires CF.

Il reste la traduction proprement dite. Une phrase n'a pas nécessairement la même structure dans les deux langues. Il faut donc transformer la structure de la langue de départ à celle (généralement moins riche car l'ordinateur "compose" ses phrases d'une façon "standard") de la langue d'arrivée par une manipulation d'arbre qui est fonction de la grammaire des deux langues. Les linguistes parlent aussi d'une structure profonde qu'ils ne peuvent encore situer précisément.

Voici donc un aperçu des méthodes utilisées pour faire de la traduction automatique. On utilise actuellement un programme pour traduire chaque jour La Pravda du russe à l'anglais. On est à mettre au point, à Grenoble, un autre programme de traduction que nous pourrons, s'il est prêt à temps, voir fonctionner au pavillon français de l'Expo.

Rédaction: Pierre Bouchard

---

(1) Il faut exclure ici la poésie: les ordinateurs ne sont pas encore en mesure de la traduire.